PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-053235
(43)Date of publication of application : 26.02.1999

(51)Int.Cl. G06F 12/00
G06F 3/06

(21)Application number : 09-214656
(22)Date of filing : 08.08.1997

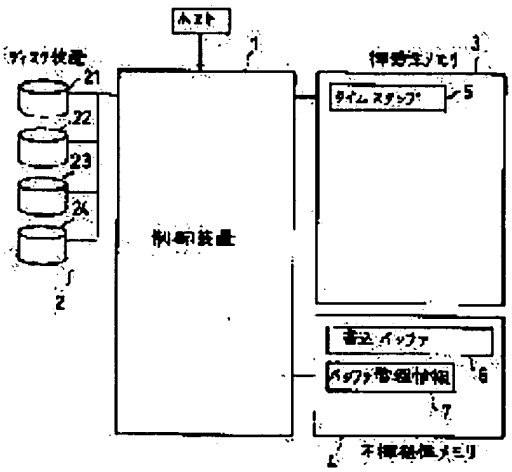
(71)Applicant : TOSHIBA CORP
(72)Inventor : SEKIDO KAZUNORI

(54) DATA UPDATING METHOD OF DISK STORAGE DEVICE AND DISK STORAGE CONTROL SYSTEM

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a data updating method for a disk storage device which does not need an indirect map in principle and is inexpensive and fast and also to construct a disk storage control system which realizes the method.

SOLUTION: This disk storage device which consists of N disk drives is provided with a write buffer 6 which has a capacity that corresponds to N×K (integer) logical blocks and accumulates a logical block of data to be updated in the write buffer. A controller 1 delays data updating of the logical block until the accumulated logical blocks reach N×K-1, generates a logical address tag block that consists of logical addresses for each logical block accumulated in the write buffer, and continuously and sequentially writes N×K logical blocks which add the logical address tag blocks to N×K-1 logical blocks in an empty that is different from an area that holds data to be updated on N disk devices.



(19)日本国特許庁（ＪＰ）　(12) **公開特許公報**（Ａ）　(11)特許出願公開番号

特開平11－53235

(43)公開日　平成11年(1999)２月26日

(51)Int.Cl.[6]	識別記号	ＦＩ	
Ｇ０６Ｆ　12/00	５１４	Ｇ０６Ｆ　12/00	５１４Ａ
3/06	５４０	3/06	５４０

審査請求　未請求　請求項の数26　ＯＬ　（全 13 頁）

(21)出願番号　特願平9－214656

(22)出願日　平成９年(1997)８月８日

(71)出願人　000003078
株式会社東芝
神奈川県川崎市幸区堀川町72番地

(72)発明者　関戸　一紀
東京都青梅市末広町２丁目９番地　株式会社東芝青梅工場内

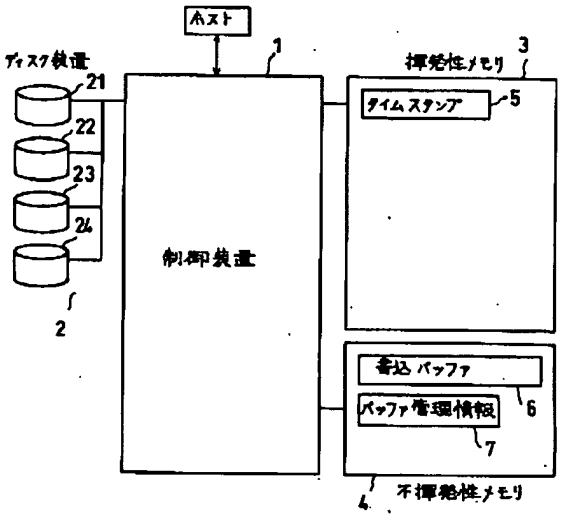
(74)代理人　弁理士　大胡　典夫　（外１名）

(54)【発明の名称】　ディスク記憶装置のデータ更新方法、ならびにディスク記憶制御システム

(57)【要約】

【課題】　本発明は、間接マップを原理的に必要としない、安価で高速なディスク記憶装置のデータ更新方法を提案するものであり、併せて同方法を実現するディスク記憶制御システムを構築することを課題とする。

【解決手段】　Ｎ台のディスク装置から構成されるディスク記憶装置において、Ｎ×Ｋ（整数）個の論理ブロックに相当する容量を持つ書込みバッファ６を備え、この書込みバッファに更新すべきデータの論理ブロックを蓄積し、制御装置１は、その蓄積した論理ブロックがＮ×Ｋ－１個に達するまでその論理ブロックのデータ更新を遅延させ、書込みバッファに蓄積された各論理ブロックに対する論理アドレスから構成される論理アドレスタグブロックを生成して、Ｎ×Ｋ－１個の論理ブロックに前記論理アドレスタグブロックを加えたＮ×Ｋ個の論理ブロックを、Ｎ台のディスク装置上の更新されるべきデータを保持している領域とは別の空領域に連続して順次書込む。



1

【特許請求の範囲】

【請求項１】　Ｎ台のディスク装置から構成されるディスク記憶装置において、Ｎ×Ｋ（整数）個の論理ブロックに相当する容量を持つ書込みバッファを有し、この書込みバッファに更新すべきデータの論理ブロックを蓄積し、その蓄積した論理ブロックがＮ×Ｋ－１個に達するまでその論理ブロックのデータ更新を遅延させ、前記書込みバッファに蓄積された各論理ブロックに対する論理アドレスから構成される論理アドレスタグブロックを生成して、Ｎ×Ｋ－１個の論理ブロックに前記論理アドレスタグブロックを加えたＮ×Ｋ個の論理ブロックを、Ｎ台のディスク装置上の更新されるべきデータを保持している領域とは別の空領域に連続して順次書込むことを特徴とするディスク記憶装置のデータ更新方法。

【請求項２】　上記書込みは、複数のディスク記装置に上記物理ブロックが並行して書込まれることを特徴とする請求項１記載のディスク記憶装置のデータ更新方法。

【請求項３】　上記論理アドレスタグブロックに書き込み順序を判定するために用いられるタイムスタンプを付加することを特徴とする請求項１記載のディスク記憶装置のデータ更新方法。

【請求項４】　ディスク装置に記録された論理アドレスタグブロックを検査し、各論理アドレスに対応するディスク装置上の位置を見つけることを特徴とする請求項１記載のディスク記憶装置のデータ更新方法。

【請求項５】　上記の検査において、同じ論理アドレスを含むストライプ（Ｎ×Ｋ個の論理ブロック）が複数ある場合、タイムスタンプが最新のストライプのブロックを有効ブロックとし、他の同論理アドレスを持つブロックを無効ブロックと判定することを特徴とする請求項４記載のディスク記憶装置のデータ更新方法。

【請求項６】　上記の検査において、最大のタイムスタンプ値を見つけ、次の書込みで付加するタイムスタンプを再生することを特徴とする請求項４記載のディスク記憶装置のデータ更新方法。

【請求項７】　上記の検査において、最小のタイムスタンプ値を見つけ、書込み順序の判定基準となるタイムスタンプ値を求めることを特徴とする請求項４記載のディスク記憶装置のデータ更新方法。

【請求項８】　上記ディスク装置上に論理ブロックを連続して書込めるような空領域を作るため、複数ストライプを読み出して有効ブロックだけを前記書込みバッファに移し、対応する論理タグブロック内の論理アドレスから新しい論理アドレスタグブロックを生成し、書込みバッファの有効データと生成された論理アドレスタグから構成されるストライプを、読み出したストライプを保持していた領域とは別の空領域に順次書込むことを特徴とする請求項４記載のディスク記憶装置のデータ更新方法。

【請求項９】　上記新しい論理アドレスタグブロックを

2

生成するとき、有効ブロック数がＮ×Ｋ－１個に満たない場合は、新しい論理アドレスタグブロック内のデータが格納されないブロックに対応した論理アドレスにはＮＵＬＬアドレスを設定することを特徴とする請求項８記載のディスク記憶装置のデータ更新方法。

【請求項１０】　上記検査において、各論理アドレスに対するストライプ番号、ストライプ内のブロック番号、有効データのタイムスタンプから構成される変換マップを生成し、ディスク装置の空領域にデータを書き込んだ後、その論理アドレスタグブロックを元に変換マップを修正し、常に有効ブロックを指すように管理することを特徴とする請求項４記載のディスク記憶装置のデータ更新方法。

【請求項１１】　変換マップ作成後、ディスク装置へのアクセスが少ない時間帯に、各ストライプの論理アドレスタグブロックを読み出して変換マップとの比較訂正を行うことを特徴とする請求項１０記載のディスク記憶装置のデータ更新方法。

【請求項１２】　論理アドレスタグブロックが記録されるディスク装置をストライプによって分散配置し、論理アドレスタグブロックの検査時、ディスク装置の読み出しを並列に行うことを特徴とする請求項４記載のディスク記憶装置のデータ更新方法。

【請求項１３】　ストライプ単位の順次書込みに加え、論理アドレスタグだけを集めた専用タグ領域にもその論理アドレスタグを書込み、論理アドレスタグブロックの検査時にはこの専用タグ領域を順次読み出して調べることを特徴とする請求項４記載のディスク記憶装置のデータ更新方法。

【請求項１４】　ディスク装置上の記憶領域をストライプ単位に複数のセグメントに分割し、一定期間には１つのセグメントにストライプが書込まれるように制御するとともに、書込むセグメントを切替えるときにはその時点の変換マップと切替え先のセグメント番号をディスク装置に記録し、変換マップ作成時には、ディスク装置のセグメント切替え時の変換マップを見込みディスク装置に記録されたセグメント番号の論理アドレスタグブロックだけを検査することを特徴とする、請求項４記載のディスク記憶装置のデータ更新方法。

【請求項１５】　不揮発性メモリ上にセグメント内の各ストライプに対応したビットマップを用意し、書込むセグメントを切替えるときにはこのビットマップをクリアし、ストライプ書込みのときには書込んだストライプに対応したビットをセットし、変換マップ作成時には、ディスク装置のセグメント切替え時の変換マップを見込み、ディスク装置に記録されたセグメント番号の論理アドレスタグの中で、ビットマップがセットされているもののみ検査することを特徴とする請求項４記載のディスク記憶装置のデータ更新方法。

【請求項１６】　タイムスタンプの最小値を進めるた

め、無効ブロックが少ないストライプに対しても定期的に、読み出して有効ブロックだけを前記書込みバッファに移し、対応する論理タグブロック内の論理アドレスと新しいタイムスタンプから論理アドレスタグブロックを生成し、書込みバッファの有効データと生成された論理アドレスタグブロックから構成されるストライプを、読み出したストライプを保持していた領域とは別の空領域に順次書込むことを特徴とする請求項７記載のディスク記憶装置のデータ更新方法。

【請求項１７】　タイムスタンプの最小値を進めるため、無効ブロックが少ないストライプに対して定期的に、論理アドレスタグブロックだけを読み出して無効ブロックの論理アドレスをＮＵＬＬアドレスとした新しいタイムスタンプを付加した論理アドレスタグブロックを生成し、ここで生成された論理アドレスタグブロックを読み出した論理アドレスタグブロックに上書きすることを特徴とする請求項４記載のディスク記憶装置のデータ更新方法。

【請求項１８】　変換マップ作成後、ディスク装置上の論理アドレスタグブロックのタイムスタンプと対応する変換マップのタイムスタンプを比較し、無効ブロックを判定することを特徴とする請求項９記載のディスク記憶装置のデータ更新方法。

【請求項１９】　Ｎ台のディスク装置から構成されるディスク記憶装置において、（Ｎ－１）×Ｋ個の論理ブロックに相当する容量を持つ書込みバッファを備え、この書込みバッファに更新すべきデータの論理ブロックを蓄積して、その蓄積した論理ブロックが選択した個数に達するまでその論理ブロックの更新を遅延させ、前記書込みバッファに蓄積された各論理ブロックに対する論理アドレスから構成される論理アドレスタグブロックを生成して、選択した個数の論理ブロックに前記論理アドレスタグブロックを加えた（Ｎ－１）×Ｋ個のデータ論理ブロックからＫ個のパリティブロックを生成し、このデータ論理ブロックにパリティブロックを加えたＮ×Ｋ個の論理ブロックをＮ台のディスク装置上の更新されるべきデータを保持している領域とは別の空領域に連続した書込み操作で順次書込むことを特徴とするディスク記憶装置のデータ更新方式。

【請求項２０】　上記選択した個数として（Ｎ－１）×Ｋ－１とし、１個のディスク装置に論理アドレスタグブロックが記録されるようにすることを特徴とする請求項１９記載のディスク記憶装置のデータ更新方法。

【請求項２１】　上記選択した個数として（Ｎ－１）×Ｋ－２とし、１パリティストライプで２個のディスク装置に論理アドレスタグブロックが記録されるように、２つの論理アドレスタブブロックを割り付けることを特徴とする請求項１９記載のディスク記憶装置のデータ更新方法。

【請求項２２】　ディスク装置に記録された論理アドレスタグブロックを検査するのに、パリティストライプ単位の順次書込みに加え、論理アドレスタグを集めた専用タグ領域にもその論理アドレスタグを書込み、この専用タグ領域の書込みデータはパリティで保護しないかわりに、パリティストライプ内の論理アドレスタグが記録されるディスク装置と専用タグ領域の論理アドレスタグが記録されるディスク装置が異なるように専用タグ領域を割り付けることを特徴とする請求項２０記載のディスク記憶装置のデータ更新方法。

【請求項２３】　Ｎ台のディスク装置から構成されるディスク記憶装置と、Ｎ×Ｋ（整数）個の論理ブロックに相当する容量を持つ書込みバッファと、この書込みバッファに更新すべきデータの論理ブロックを蓄積し、その蓄積した論理ブロックがＮ×Ｋ－１個に達するまでその論理ブロックのデータ更新を遅延させ、前記書込みバッファに蓄積された各論理ブロックに対する論理アドレスから構成される論理アドレスタグブロックを生成して、Ｎ×Ｋ－１個の論理ブロックに前記論理アドレスタグブロックを加えたＮ×Ｋ個の論理ブロックを、Ｎ台のディスク装置上の更新されるべきデータを保持している領域とは別の空領域に連続して順次書込む制御装置とを具備することを特徴とするディスク記憶制御システム。

【請求項２４】　書き込みの時間的順序を維持するタイムスタンプが格納される揮発性メモリと、ディスク装置に書き込むべきデータをログ構造化して保持する上記書き込みバッファ、ならびに、書き込みバッファ内の空き領域及び保持されている書き込みデータの論理アドレス情報を保持するバッファ管理情報が格納される不揮発性メモリを具備することを特徴とする請求項２３記載のディスク記憶制御システム。

【請求項２５】　Ｎ台のディスク装置から構成されるディスク記憶装置と、（Ｎ－１）×Ｋ個の論理ブロックに相当する容量を持つ書込みバッファと、この書込みバッファに更新すべきデータの論理ブロックを蓄積して、その蓄積した論理ブロックが選択した個数に達するまでその論理ブロックの更新を遅延させ、前記書込みバッファに蓄積された各論理ブロックに対する論理アドレスから構成される論理アドレスタグブロックを生成して、選択した個数の論理ブロックに前記論理アドレスタグブロックを加えた（Ｎ－１）×Ｋ個のデータ論理ブロックからＫ個のパリティブロックを生成し、このデータ論理ブロックにパリティブロックを加えたＮ×Ｋ個の論理ブロックをＮ台のディスク装置上の更新されるべきデータを保持している領域とは別の空領域に連続した書込み操作で順次書込制御装置とを具備することを特徴とするディスク記憶制御システム。

【請求項２６】　パリティを使った冗長性ディスク構成をとるために冗長ディスク装置を付加し、更に、書き込みの時間的順序を維持するタイムスタンプが格納される揮発性メモリと、ディスク装置に書き込むべきデータを

ログ構造化して保持する上記書き込みバッファ、ならびに、書き込みバッファ内の空き領域及び保持されている書き込みデータの論理アドレス情報を保持するバッファ管理情報が格納される不揮発性メモリを具備することを特徴とする請求項２５記載のディスク記憶制御システム。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、特に、ＲＡＩＤ（Redundant Array of Inexpensive Disk）等ディスクアレイの入出力に用いて好適な、ディスク記憶装置のデータ更新方法、ならびにディスク記憶制御システムに関する。

【０００２】

【従来の技術】米国特許第５，１２４，９８７号、特開平６－２１４７２０号、特開平６－２６６５１０号に開示されているように、ディスク記憶装置への高速な書込み方法として、旧データの領域を書き換えるのでは無く、更新データを溜めておき、ディスク装置内のあらかじめ用意した別の空領域にまとめて書込む方法が提案されている。

【０００３】図１７を用い、上述した従来の方法について簡単に説明する。図において、論理ブロックL6，L4，L2，L12,L7,L11を更新する場合を考える。これらの論理ブロックの旧データはディスク装置内のP6,P4,P2,P12,P7,P11 に存在する。よって、このP6,P5,P2,P12の内容を更新するのが普通であるが、この方法ではP6,P4,P2,P12,P7,P11 のデータはそのままにして、新しい論理ブロックL6,L４,L2,L12,L7,L11のデータをあらかじめ用意した別の空領域であるP51,P52,P53,P54,P55,P56 にまとめて書込む。これにより、６回の書込み操作が３回の書込み操作に減り、書込み性能の向上になる。

【０００４】尚、図１７に示す例では説明を簡単にするため、１ディスクに２ブロックしか書かれていないが、実際には数＋ブロックがまとめて書込まれる。また、ＲＡＩＤ４、５（いずれもアクセス耐力を向上させる手法であり、ＲＡＩＤ４；データの配置単位をビットやバイトのような小さな単位ではなく、セクタやブロックのような大きな単位とし、小容量のデータ読み出し要求に対してディスクを独立動作可とする方式。ＲＡＩＤ５；冗長データを専用のパリティディスクに格納するのではなく、各データディスクに巡回的に配置する方式）では１個のストライプ丸ごとの書き換えになるためパリティ維持のためのディスク読み出しも不要になり、書込み時のオーバヘッド減少にもなる。

【０００５】一方、論理ブロックL6,L5,L2,L12,L7,L11の最新データはディスク装置内のP51,P52,P53,P54,P55,P56 に存在するので、間接マップの内容を正しいディスク位置を指すように書き換える。また、論理ブロックのデータを読み出すときには、この間接マップを調べて最新の位置を求めてから読み出すので、旧データを読み出す危険性はない。

【０００６】

【発明が解決しようとする課題】上述した従来技術においては、間接マップにより最新データの位置情報を管理していたため、間接マップの障害や誤操作によりそのデータが無くなると、ディスク装置内の全データの喪失になるというデータ保全性の問題があった。また、全論理ブロックに対して間接マップを用意する必要があり大容量、かつ、電源障害に備え間接マップを保持するのに不揮発性メモリが必要であったため、間接マップが非常に高価になるという問題もあった。

【０００７】本発明は上記の問題を解決するためになされたものであり、間接マップを原理的に必要としない、安価で高速なディスク記憶装置のデータ更新方法、ならびにディスク記憶制御システムを提供することを目的とする。

【０００８】

【課題を解決するための手段】本発明のディスク記憶装置のデータ更新方法は、Ｎ台のディスク装置から構成されるディスク記憶装置において、Ｎ×Ｋ（整数）個の論理ブロックに相当する容量を持つ書込みバッファを有し、この書込みバッファに更新すべきデータの論理ブロックを蓄積し、その蓄積した論理ブロックがＮ×Ｋ－１個に達するまでその論理ブロックのデータ更新を遅延させ、前記書込みバッファに蓄積された各論理ブロックに対する論理アドレスから構成される論理アドレスタグブロックを生成して、Ｎ×Ｋ－１個の論理ブロックに前記論理アドレスタグブロックを加えたＮ×Ｋ個の論理ブロックを、Ｎ台のディスク装置上の更新されるべきデータを保持している領域とは別の空領域に連続して順次書込むことを特徴とする。また、Ｎ台のディスク装置から構成されるディスク記憶装置において、（Ｎ－１）×Ｋ個の論理ブロックに相当する容量を持つ書込みバッファを備え、この書込みバッファに更新すべきデータの論理ブロックを蓄積して、その蓄積した論理ブロックが選択した個数に達するまでその論理ブロックの更新を遅延させ、前記書込みバッファに蓄積された各論理ブロックに対する論理アドレスから構成される論理アドレスタグブロックを生成して、選択した個数の論理ブロックに前記論理アドレスタグブロックを加えた（Ｎ－１）×Ｋ個のデータ論理ブロックからＫ個のパリティブロックを生成し、このデータ論理ブロックにパリティブロックを加えたＮ×Ｋ個の論理ブロックをＮ台のディスク装置上の更新されるべきデータを保持している領域とは別の空領域に連続した書込み操作で順次書込むことを特徴とする。

【０００９】本発明のディスク記憶制御システムは、Ｎ台のディスク装置から構成されるディスク記憶装置と、Ｎ×Ｋ（整数）個の論理ブロックに相当する容量を持つ書込みバッファと、この書込みバッファに更新すべきデ

ータの論理ブロックを蓄積し、その蓄積した論理ブロックがＮ×Ｋ－１個に達するまでその論理ブロックのデータ更新を遅延させ、前記書込みバッファに蓄積された各論理ブロックに対する論理アドレスから構成される論理アドレスタグブロックを生成して、Ｎ×Ｋ－１個の論理ブロックに前記論理アドレスタグブロックを加えたＮ×Ｋ個の論理ブロックを、Ｎ台のディスク装置上の更新されるべきデータを保持している領域とは別の空領域に連続して順次書込む制御装置とを具備することを特徴とする。また、Ｎ台のディスク装置から構成されるディスク記憶装置と、（Ｎ－１）×Ｋ個の論理ブロックに相当する容量を持つ書込みバッファと、この書込みバッファに更新すべきデータの論理ブロックを蓄積して、その蓄積した論理ブロックが選択した個数に達するまでその論理ブロックの更新を遅延させ、前記書込みバッファに蓄積された各論理ブロックに対する論理アドレスから構成される論理アドレスタグブロックを生成して、選択した個数の論理ブロックに前記論理アドレスタグブロックを加えた（Ｎ－１）×Ｋ個のデータ論理ブロックからＫ個のパリティブロックを生成し、このデータ論理ブロックにパリティブロックを加えたＮ×Ｋ個の論理ブロックをＮ台のディスク装置上の更新されるべきデータを保持している領域とは別の空領域に連続した書込み操作で順次書込制御装置とを具備することを特徴とする。

【００１０】上記した構成をとることにより、間接マップを原理的に必要としない、安価で高速なディスク記憶装置、ならびにディスク記憶制御システムを構築できる。

【００１１】

【発明の実施の形態】図１は本発明を適用して構成したディスク記憶装置の概念構成図である。本発明のディスク記憶制御システムは、制御装置１、ディスク装置２（２１，２２，２３，２４）、揮発性メモリ３、不揮発性メモリ４から構成される。揮発性メモリ３には、書込の時間的順序を維持するためのタイムスタンプ５、不揮発性メモリ４には、ディスク装置２に書き込むデータをログ構造化して保持する書込バッファ６、書込バッファ６内の空き領域および保持されている書込データの論理アドレスの情報を保持するバッファ管理情報７が格納されている。制御装置１はこれらタイムスタンプ５、書込バッファ６、バッファ管理情報７を管理し、ディスク装置２への書込みを制御する。

【００１２】図２に、不揮発性メモリ４に割り付けられる書込みバッファ６とバッファ管理情報７の関係を示す。制御装置１は外部接続されるホスト機器から要求された書込みデータを、ディスク装置２に対して即書込まずに、ブロック単位に分割して書込みバッファ６に順番に詰めて（ログ形式に）格納していく。この時、書込みデータのホスト機器から見た論理アドレスを、バッファ管理テーブル７の格納したバッファ領域に対応するエントリーに保存する。また、そのエントリにデータが割り当てられたことを示すフラッグ“Ｆ”を立てる。よって、このバッファ管理情報（テーブル）７を調べることにより、ホスト機器から受け取った書込みデータを格納すべき次のバッファ領域を決めることが出来る。図２に示す例では、バッファ領域Ｂ７まで書込みデータが格納されており、Ｂ０、Ｂ１，...、Ｂ７の論理アドレスがＬＡ１３４、ＬＡ９９、...、ＬＡ６７８であることを表わしている。

【００１３】また、ディスク装置２（２１～２４）は、それぞれブロックサイズの整数倍（Ｋ）であるストライプユニットと呼ぶあらかじめ決められた単位（そのディスク装置の１トラック長に近いサイズが良い。）で書込みを行う。この時、ディスク装置２１～２４の物理的に同じ位置のストライプユニットは１つのストライプとして、同じタイミングで書込みが行われる。また、実際のディスク装置２１～２４を合わせて全記憶容量よりも少ない容量のディスク装置としてホスト機器に見せている。具体的には、ホスト機器が最初に記憶容量を問い合わせて来たとき、その返答として少ない容量を返す。従って、ホスト機器から論理的に読み書きできる記憶領域のほかに余分な記憶領域が確保されることになる。この領域を空領域と呼ぶことにする。

【００１４】更に、タイムスタンプ５は、ホストからの書込みデータが実際にディスク装置２に書込まれる時に付加される情報で、ディスク装置２内でのデータ書込み順序を判定するのに用いる。よって、書込みバッファ６のデータがディスク装置２に書込まれる毎にタイムスタンプ５がインクリメントされる。

【００１５】図１に示す本発明実施例の動作について図２～図１２を参照しながら詳細に説明する。

【００１６】まず、書込み動作から説明する。ホスト機器から書込むべきデータとその論理アドレスを受け取った制御装置１は、不揮発性メモリ４上の書込みバッファ６の空領域にブロック単位に分割して詰めて格納する。また、受け取った論理アドレスはブロック毎のアドレスに変換して、バッファ管理テーブル７の対応するエントリに格納する。なお、既に書込みバッファ６に格納されているデータに対する更新データの場合には、書込みバッファ６の空領域に詰めて格納するのではなく、直接書込みバッファ６内の旧データを変更する。

【００１７】ホスト機器からの書込みデータが１ストライプ分に１ブロック少ない数（Ｋ＊４－１）だけ書込みバッファ６に溜まった段階で、制御装置１はそれらのデータをディスク装置２に書込みに行く。この時、最後の書込みブロックとして、書込み管理テーブル７に格納された各ブロックの論理アドレスと揮発性メモリ３上のタイムスタンプ５から論理アドレスタグブロックを作成する。この論理アドレスタグブロック内のアドレスデータとデータブロックの間には、１対１の関係があらかじめ

設定されており、各データブロックの論理アドレスが分かるようになっている。その後、この論理アドレスタグブロックを加えた１ストライプ分のデータを、まとめてディスク装置２１～２４の空領域に同時に書込む。この様子は図３に示されている。

【００１８】また、タイムスタンプ５の値は書込みが完了した段階でインクリメントされる。このように、多数の細かいディスク書込みを１回にまとめられるので、ディスク書込み性能が大きく向上する。

【００１９】次に、詰替え処理について説明する。旧データの領域を直接書き換えるのではなく、更新データを溜めておき、ディスク装置２内のあらかじめ用意した別の空領域にまとめて書込む方法では空領域が常に存在することが必須である。そのため、ホスト機器からのディスクアクセスが空いている間に、既に他の領域にデータが書込まれ無効になっているデータを寄せ集めて空領域を作る必要がある。この処理を詰替え処理と呼ぶ。この詰替え処理は無効ブロック判定とストライプ統合の２つのステップからなる。

【００２０】無効ブロック判定の例として、図４に示す順番でホスト機器から１ブロックサイズのデータ書込みがある場合を考える。図中のＬ××はホストから渡される論理アドレス、Ｓ××は書込み順番を表わす。本発明実施例では、書込みバッファ６は１５ブロックのデータを保持できるので、最初のＳ１～Ｓ１５の書込みデータが１つのストライプ（ＳＴ１）にまとめられ、タイムスタンプＴ１が付加されてディスク装置の空領域に書き出される。同様に、Ｓ１６～Ｓ３０の書込みデータが別のストライプ（ＳＴ２）としてタイムスタンプＴ２が付加されて別の空領域に書き出される。なお、書き出し毎にタイムスタンプ５はインクリメントされるのでＴ１＜Ｔ２の関係がある。

【００２１】ここで、図から分かるように論理アドレスＬ９、Ｌ１８のデータはタイムスタンプＴ１のストライプではＳ５、Ｓ２として、タイムスタンプＴ２のストライプではＳ１９、Ｓ２１のブロックとして重複して存在する。書込まれた順番を考えると、Ｓ１９，Ｓ２１のデータブロックが有効であり、Ｓ５、Ｓ２のデータは無効と判定されなければならない。しかし、ここで便宜上使った書込み順番Ｓ××は実際のディスク上に記録されていない。

【００２２】そこで、ストライプ内の論理アドレスタグを使ってこの判定を行う。図４の例における、２つのストライプＳＴ１，ＳＴ２の論理アドレスタグＴＧ１、ＴＧ２の内容は図５の通りである。図から分かるように、２つの論理アドレスタグＴＧ１、ＴＧ２に同じ論理アドレスＬ９、Ｌ１８のデータが含まれており、ストライプＳＴ１のブロックＢ５、Ｂ２とストライプＳＴ２のブロックＢ４、Ｂ６のどちらかのデータが無効である。さらに、論理アドレスタグＴＧ１のタイムスタンプＴ１と論理アドレスタグＴＧ２のタイムスタンプＴ２を比較すると、Ｔ１＜Ｔ２の関係にあることから、ストライプＳＴ１のブロックＢ５、Ｂ２が無効であることが判定できる。以上説明のように、ディスク装置２内の論理アドレスタグを調べることにより無効なデータブロックを見つけることができる。

【００２３】ストライプ統合の例として、図６に示すように２つのストライプＳＴ３，ＳＴ４を１つのストライプＳＴ５に統合する場合を考える。図に示すように、ストライプＳＴ３ではＢ２、Ｂ７，Ｂ８，Ｂ１２，Ｂ１３の５ブロックが有効で他の１０ブロックは無効（ハッチング）であるとする。同様に、ストライプＳＴ４ではブロックＢ１８，Ｂ１９，Ｂ２０，Ｂ２１，Ｂ２２，Ｂ２４，Ｂ２５、Ｂ２７，Ｂ２９の９ブロックが有効で他の６ブロックが無効（ハッチング）であるとする。２つのストライプの有効ブロックは合わせて１４ブロックしかないので、この２つのブロックを１つに統合することにより、結果として１つの空領域が作れる。

【００２４】ストライプ統合では図６に示すように、２つのストライプＳＴ３，ＳＴ４を揮発性メモリ３内に読み出し、有効ブロックだけを詰めて書込みバッファ６に移す。それに合わせ、図７に示すように、論理アドレスタグもＴＧ３、ＴＧ４から有効ブロックの論理アドレスだけを対応する位置に移し、新しい論理アドレスタグＴＧ５を作りその時点のタイムスタンプを更新する。

【００２５】この例では１４個の有効ブロックしかなかったので、更にホスト機器から１つの書込みブロックを待ってストライプを完成させてディスク装置２の空領域にまとめて書込む。この場合、ディスク領域は有効に活用されるが、ホスト機器からバーストでディスクアクセスがある場合、書込みを待たすためディスクアクセスを集中させてしまう危険性がある。そこで、最後のデータブロックは空状態のままアクセスが空いている間に書込んでしまうことも可能である。このとき、論理アドレスタグＴＧ５の最後のデータブロックに対する論理アドレスには－１等ＮＵＬＬアドレスを入れることによりデータが入っていないことを表わせるので問題はない。

【００２６】次に、このように書かれたデータブロックの読み出し動作について説明する。詰め替え処理の無効ブロック判定を、ディスク装置２内の全ストライプの論理アドレスタグに対して行うことにより、全論理アドレスに対する有効ブロックの物理的位置を検出できる。従って、原理的には、ホスト機器から読み出しブロックの論理アドレスを受け取る度に全ストライプのチェックを行うことにより、読み出すべき物理ブロックを見つけ出すことが出来る。しかし、この方法ではブロック読み出しに膨大な時間が掛ってしまい実用的でない。

【００２７】そこで、システム起動時にだけ全ストライプの論理アドレスタグの調査を行ない、揮発性メモリ３上に論理アドレスから物理アドレスへの変換マップを作

る。ホスト機器からの読み出し要求に対してはこの変換マップを使って有効ブロックへのアクセスを行う。これにより、常にアドレスタグの調査をしなくても良く、読み出し時に性能が大きく低下することはない。また、この変換マップは何時でも全ストライプを調査することで再生できるため、従来のように電源障害に備えて不揮発メモリ4に格納する必要もない。

【0028】ここで、変換マップについて図8を用いて説明する。変換マップは図に示すように、各論理アドレスに対するブロックが格納されているストライプ番号ST#とそのストライプ内のブロック番号BLK#、さらにそのタイムスタンプTS#をテーブル形式で保持している。従って、論理アドレスL0～Lnが与えられれば、このテーブルを索引することによりST#とBLK#から簡単に実際の物理アドレスが求まる。

【0029】また、システム起動時の変換マップの作成は、調査した論理アドレスタグの全論理アドレスについて、テーブルのタイムスタンプより論理アドレスタグのタイムスタンプが大きいときだけ、そのストライプ番号と対応するブロック番号をテーブルに登録する。この調査を全ストライプについて行えば、有効ブロックだけを指す変換マップが出来る。更に、ディスク装置2にストライプを書込む毎に、その論理アドレスタグに対して同様の処理を行うことにより、この変換マップは常に有効なブロックを指す。また、ディスクアクセスが空いているときに、各ストライプの論理アドレスタグと変換マップを比較検査することにより、メモリ障害等でこの変換マップが不正な値になっても検出訂正が可能である。

【0030】以上説明のように、変換マップ作成の主な処理は論理アドレスタグの検査である。故に、大容量ディスク装置のように論理アドレスタグ数が多い場合、電源障害やシステム起動時の変換マップ作成に長時間かかってしまう。特に、図2に示すように、論理アドレスタグブロックが1台のディスク装置24に集中すると、システム起動時にはこのディスクにアクセスが集中し、論理アドレスタグの調査を並列に行うことが出来ない。そこで、図9に示すように、ストライプによって論理アドレスタグが格納されるディスク装置を4台に分散し並列に論理アドレスタグを調査することにより、この変換マップ作成に要する時間を1／4に短縮できる。

【0031】この他、ディスク装置2の記憶領域を複数のセグメントに分割管理することにより、変換マップ作成に必要な論理アドレスタグの検査数を削減できる。図10にセグメント分割方式におけるディスク装置の記憶領域の構成を示す。図に示すように、ディスク装置の記憶領域は、ストライプを単位としてセグメント管理情報（ハッチング）と4つのセグメントに分割される。ここで、セグメントとは書込みバッファデータの一括書込みや詰め替え処理のディスク書込みがある期間集中して行われる単位領域のことである。例えば、セグメント2がディスク書込みの対象である間は、セグメント1、3、4には書込みに行かないように空領域の選択を制御する。

【0032】また、あるセグメントの空領域が少なくなりディスク書込みを他のセグメントへ切替えるときにはセグメント管理情報をディスク装置に保存する。セグメント管理情報は図11に示すように、セグメント番号と切替え時変換マップから構成される。セグメント番号とは切替え先のセグメント番号で、切替え時変換マップとはセグメントを切替える時点の揮発性メモリ3上の変換マップの状態である。

【0033】なお、切替え時変換マップはセグメントが切替える度に全て上書きするのではなく、直前のセグメントに書込まれた論理アドレスのエントリだけを書き戻せばよい。従って、前回のセグメント切替え時にタイムスタンプを覚えておき、変換マップのタイムスタンプを比較することにより、直前のセグメントに書込まれた論理アドレスを判定できる。

【0034】このセグメント分割方式では、セグメント切替え時にセグメント管理情報を保存している。よって、セグメント切替え時の変換マップをセグメント管理情報から読み出して、その後でセグメント管理情報のセグメント番号で指されるセグメントの論理アドレスタグだけを検査するだけで、全論理アドレスタグを検査した場合と同じ変換マップが再現できる。従って、この方式により必要な論理アドレスタグの検査数は1セグメント分で良く、この例では変換マップの作成に要する時間を1／4に短縮できる。

【0035】更に、不揮発性メモリ4上にセグメント内の全ストライプに対応したビットマップを用意して、セグメント切替え時にはこのビットマップをクリアし、一括書込みや詰替えの書込み時には書込んだストライプに対応するビットを“1”にセットする。これによりセグメントを切替えてから変更の有ったストライプだけがビットマップが“1”になる。従って、変換マップ作成時にこのビットマップを参照し、変更の有ったストライプの論理アドレスタグだけを検査することで検査数をさらに減らせ、変換マップ作成に要する時間を更に短縮できる。

【0036】通常論理アドレスタグのサイズは512～1024バイトであり、ディスクのシーケンシャルアクセスとランダムアクセスに約50倍の性能差がある。図2に示す方式では論理アドレスタグの情報が各ストライプ毎とびとびに存在するので、変換マップの作成では時間のかかるランダムアクセスを行っていた。そこで、図12に示すように、論理アドレスタグだけを連続して格納する専用タグ領域を（セグメント分割する場合は、各セグメント毎に）用意し、50倍も高速なシーケンシャルアクセスで論理アドレスタグを読み出せるようにする。

【００３７】そして、ホスト機器からのデータを一括して書込みや詰め換えデータの書込み時には、空領域だけでなく対応する専用タグ領域にも論理アドレスタグを書込むようにする。この方法では図２の方式では１ストライプ当たり４回のディスク書込みだったのが、専用領域への論理アドレスタグの書込みのため１回増える。しかし、変換マップ作成が５０倍も高速になるので、ディスク装置の立ち上がり時間が問題となるときには非常に有効な手段である。専用タグ領域への書込み時間を最小にするため、専用タグ領域は図１２に示すように対象領域の中心してシーク時間を減らす。また、ディスク装置２はセクタ（５１２バイトなど）単位の書込みであり、専用タグ領域内の論理アドレスタグはセクタ単位で割り付け論理アドレスタグの書込み時に読み出しが必要ないようにする。

【００３８】最後に、タイムスタンプについて説明する。図１に示すように、タイムスタンプは揮発メモリ３上に記憶されているので、電源障害などにより揮発メモリ３上のタイムスタンプが無くなってしまう。そこで、変換マップと同様にして、システム起動時にだけ全ストライプの論理アドレスタグを調査し、一番大きなタイムスタンプ５の次の値を揮発メモリ３上のタイムスタンプ５にセットする。なお、変換マップ作成の説明で述べた時間短縮手法がそのままタイムスタンプの再生にも適用できる。

【００３９】また、タイムスタンプ５はディスク装置に書込むごとにインクリメントされディスク上の書込み順序の判定するのみに使われる。例として、タイムスタンプ５が２４ビットのカウンタで構成される場合で説明する。２４ビットカウンタでは、１６M回の書込みでカウンタが一周してゼロに戻ってしまう。そこで、一般的には有効なタイムスタンプの最小値を基準にそれより小さい値は１６Mを加えて比較して判定する。この最小値も同様にシステム起動時にだけ全ストライプの論理アドレスタグを調査して求める。

【００４０】しかし、この手法が使えるのはタイムスタンプの最大値が最小値を追い越さないこと、つまり、タイムスタンプの最大値と最小値の差が２４ビットで表わせる範囲以内であることを前提にしている。よって、タイムスタンプ５が一周前に必ず全ストライプを更新してタイムスタンプ値を新しく更新する必要がある。これには、無効ブロックが少なくても予め設定した書込み回数の間更新されなかったストライプを詰替えの対象として選ぶように制御するか、無効ブロックの論理アドレスをＮＵＬＬアドレスにした、そのストライプの論理アドレスタグだけを書き換える。ＮＵＬＬアドレスを使う方法は論理アドレスタグブロックの書き換えであるので詰替えに比べて非常に軽い処理である。

【００４１】尚、上述した実施例では、無効ブロックの判定に２つストライプＳＴ１，ＳＴ２の論理アドレスタグを相互に比較して判定する方法のみ説明したが、全無効ブロックを調べるには２つのストライプ間の全組み合わせを調べなければならない。しかし、変換マップがあれば論理アドレスタグ内の各論理アドレスについて、有効データを指している変換マップのタイムスタンプとそのストライプのタイムスタンプを比較し、値が小さいブロックを無効ブロックと判定できる。

【００４２】図１はデータを複数ディスクに分散するＲＡＩＤ０の構成を示したが、本発明方式は、パリティを使った冗長性ディスク構成（ＲＡＩＤ４、５）の場合にも適用できる。図１３に本発明を適用して構成したＲＡＩＤ５構成のディスク記憶装置の概念図を示す。図１の構成に冗長用のディスク装置２５が追加された構成であり、制御装置１、ディスク装置２（２１，２２，２３，２４）、揮発性メモリ３、不揮発性メモリ４、タイムスタンプ５、書込バッファ６、バッファ管理情報７は図１に示す実施例と同じ機能を有する。

【００４３】図１３に示す実施例の動作につき、図１に示す実施例との差異に着目して説明を行なう。書込み処理では、ホストからの書込みデータが１ストライプ分に１ブロック少ない数（Ｋ＊４－１）だけ書込みバッファ６に溜まった段階で、制御装置１はそれらのデータをディスク装置２１～２５に書込みに行く。この時、最後の書込みブロックとして、書込み管理テーブル７に格納された各ブロックの論理アドレスと揮発性メモリ３上のタイムスタンプ５から論理アドレスタグブロックを作成するまでは図１に示す実施例と同じである。

【００４４】その後、この論理アドレスタグブロックを加えた１ストライプ分のデータからストライプユニット毎の排他論理輪演算（ＸＯＲ）を行い、パリティのストライプユニットを作成する。そして、このパリティ付きのストライプのデータをまとめてディスク装置２１～２５の空領域に同時に書込む。また、タイムスタンプ５の値は書込みが完了した段階でインクリメントされる。このように多数の細かいディスク書込みを１回にまとめられるのに加え、パリティ計算に旧データや旧パリティのブロックを読む必要がないので更にディスクアクセス回数を減らすことができる。なお、ストライプ詰替え処理のディスク書込みでも同様にしてパリティ付きのストライプを作成してからディスク装置２に書込む。この様子を図１４に示す。

【００４５】パリティＲＡＩＤ構成では、１台のディスク装置が故障しても、故障したディスクのデータはストライプを構成する他ディスクのデータとパリティのＸＯＲを計算することで再現でき、ディスク記憶装置としてのサービスが継続できる。しかし、システム起動時に一台故障していた場合、論理アドレスタグを格納していないディスク装置のデータも読み出して論理アドレスタグを再生してから検査するため、変換マップ作成に時間がかかりシステム起動が完了するまでの時間が大幅に増大

してしまう。

【００４６】そこで、図１５に示すように、ストライプを構成するデータブロックを１つ減らして２つのディスク装置に同じ論理アドレスタグを書くように制御する。これにより、ディスク装置が１台故障しても変換マップ作成時には残っている方の論理アドレスタグを読み出せるので、システム起動に要する時間の大幅な増大を回避できる。

【００４７】また、変換マップ作成の高速化のために専用タグ領域を活用する場合、図１６に示されるように、論理アドレスタグが専用タグ領域に格納されるディスク装置と、ストライプに格納されるディスク装置を違うように専用タグ領域の論理アドレスタグの割り付けを制御することにより、ストライプ内の論理アドレスタグは１つでよくなる。

【００４８】尚、専用タグ領域へ論理アドレスタグを書く場合もパリティによるディスク障害対策を行うと、従来１回の書込み増で済んでいたものが、２回の書込みと２回の読み出しが必要になって一括書込み時やストライプ詰替え時のディスク書込みのオーバヘッドが大きく増大する。従って、この専用タグ領域の情報はパリティで障害対策しない。この情報は変換マップ高速化のためであり、故障したディスク装置の専用タグ領域に格納されていた論理アドレスタグは、ストライプ中のものを（ランダムアクセスで）見れば良いので問題はない。また、ランダムアクセスで検査する論理アドレスタグは１／５だけであるので、変換マップ作成の高速化の効果はある。

【００４９】本発明は旧データの領域を書き換えるのでは無く、更新データを溜めておきディスク装置内のあらかじめ用意した別の空領域にまとめて書込む方法が有効なあらゆる分野に適用できる。つまり、磁気ディスクだけでなくシーケンシャル書込みとランダム書込みで大きく性能が違う光磁気ディスク等のディスク装置や、小ブロックの更新では２読み出しと２書込みが必要なパリティによる冗長性を持たせたＲＡＩＤ構成の記憶装置に主に適用できる。

【００５０】

【発明の効果】以上説明のように本発明によれば、間接マップを原理的に何時でも再生できるので、電源障害に備えて不揮発性メモリに間接マップに保持する必要はなく、従って非常に安価なディスク記憶装置を構築できる。また、ハードウェア障害等により不揮発性メモリの内容を喪失した場合でも、従来の方法では間接マップが再生できないのでディスク上のデータ全てを失ってしまうのに対し、書込みバッファに保持されていた最近の書込みデータだけが失われるだけでほとんどのディスク上データはそのまま残っている。従って、障害に対する耐久性も大きく向上する。更に、電源障害等から回復処理と通常のシステム起動処理は全く同じで済むので、システム終了時や回復時の特別な処理が不要で開発コストを軽減できる。

【００５１】また、システム起動時の処理も、複数ディスク装置への論理アドレスタグの分散配置、論理アドレスタグを順次アクセスできる専用タグ領域、記憶領域のセグメント分割管理などにより高速化できるので、システム起動時の待ち時間を実用上問題無い範囲に押さえられる。特に、パリティＲＡＩＤ構成では、論理アドレスタグを２台のディスク装置に記録することにより、１台のディスク装置が故障してもシステム起動の時間が増加しないようにできる。

【図面の簡単な説明】

【図１】本発明の実施例を示すブロック図。

【図２】本発明実施例における書き込みバッファとバッファ管理情報の関係を示すために引用した図。

【図３】本発明実施例におけるディスク装置の空領域に格納される内容を示す図。

【図４】ホスト機器から１ブロックサイズのデータ書き込み順序を示すために引用した図。

【図５】図４の例におけるストライプＳＴ１，ＳＴ２の論理アドレスタグＴＧ１／ＴＧ２の内容を示す図。

【図６】ストライプＳＴ３／ＳＴ４を１個のストライプＳＴ５に統合する例を示す図。

【図７】ストライプ統合において、論理アドレスタグＴＧ３／ＴＧ４から論理アドレスタグＴＧ５を作場合の例を示す図。

【図８】本発明実施例において使用される変換マップの構成例を示す図。

【図９】ストライプにより論理アドレスタグが格納されるディスク装置を４台に分散配置した例を示す図。

【図１０】セグメント分割におけるディスク装置の記憶領域の割り当てを示す図。

【図１１】セグメント管理情報のエントリ構成を示す図。

【図１２】論理アドレスタグを連続して格納する専用タグ領域の内容を示す図。

【図１３】本発明を適用して構成したＲＡＩＤ５によるディスク装置の実施例を示すブロック図。

【図１４】図１３に示す実施例の動作概念を示す図。

【図１５】２個のディスク装置に同じ論理アドレスタグを書くようにして制御する例を示す図。

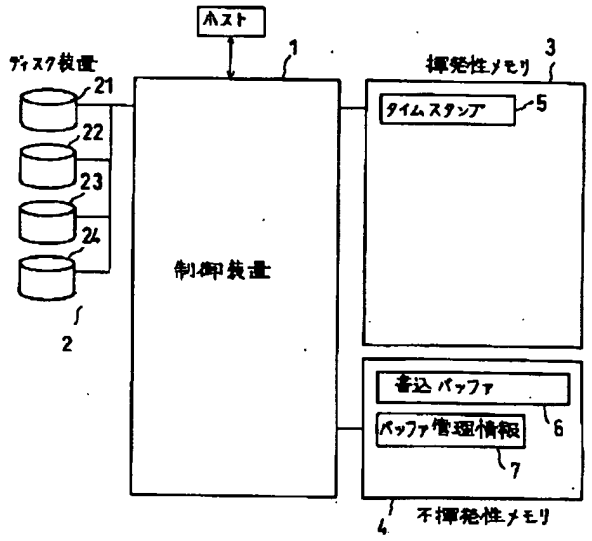
【図１６】変換マップ作成の高速化のために、専用タグ領域を割付け使用する場合の例を示す図。

【図１７】従来例におけるデータ更新方法を実現するためのシステム構成を示す図。

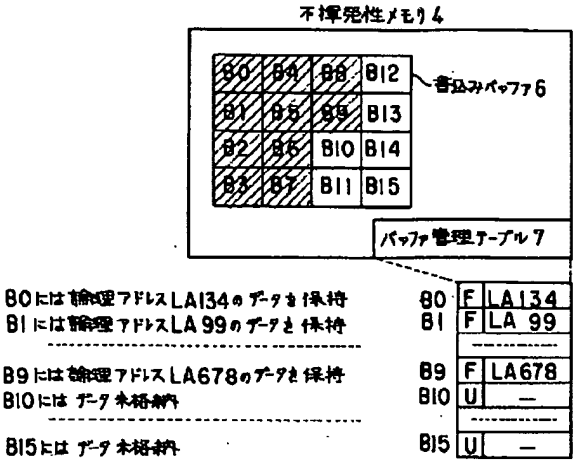
【符号の説明】

１…制御装置、２（２１，２２，２３，２４，２５）…ディスク装置、３…揮発性メモリ、４…不揮発性メモリ、５…タイムスタンプ、６…書き込みバッファ、７…バッファ管理情報（テーブル）

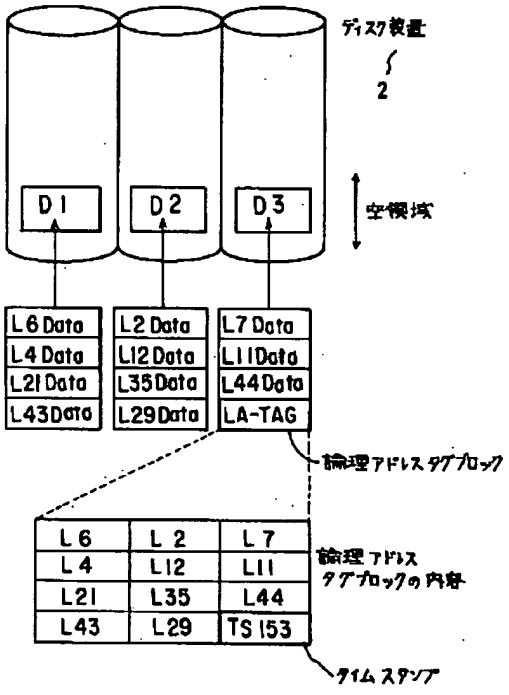
【図１】



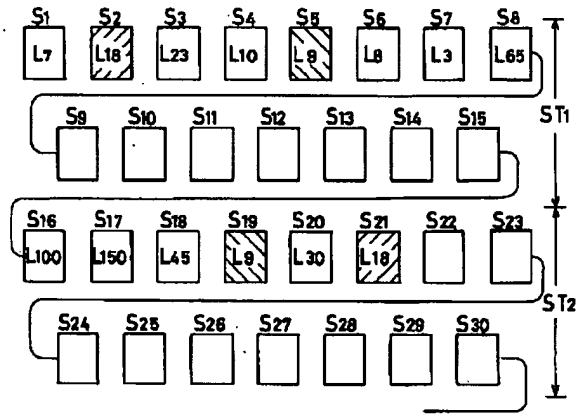
【図２】



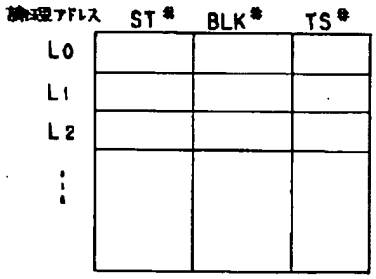
【図３】



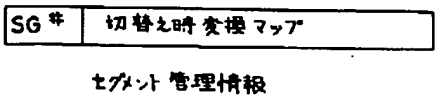
【図４】



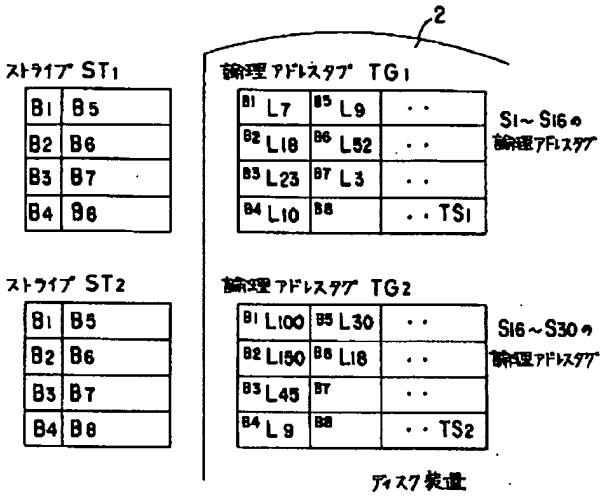
【図８】



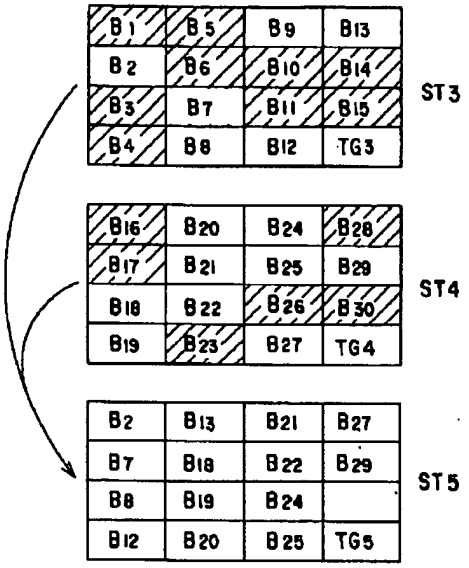
【図１１】



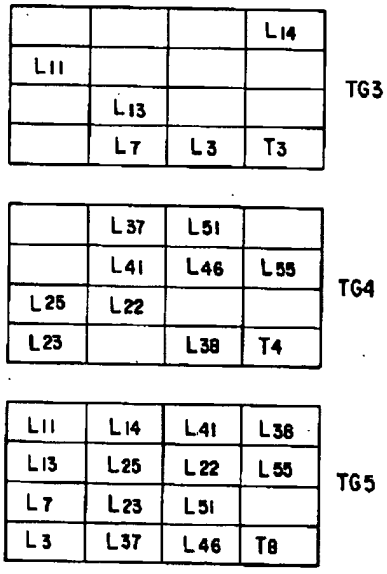
【図5】



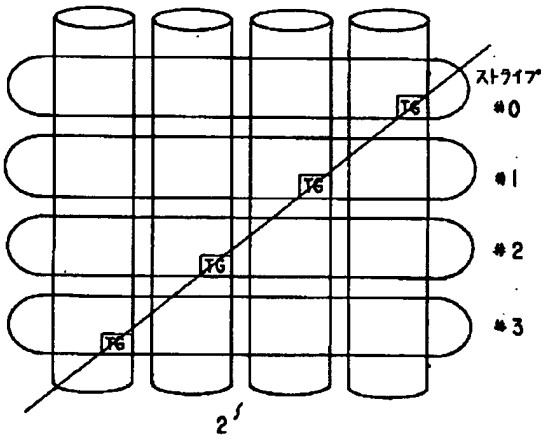
【図6】



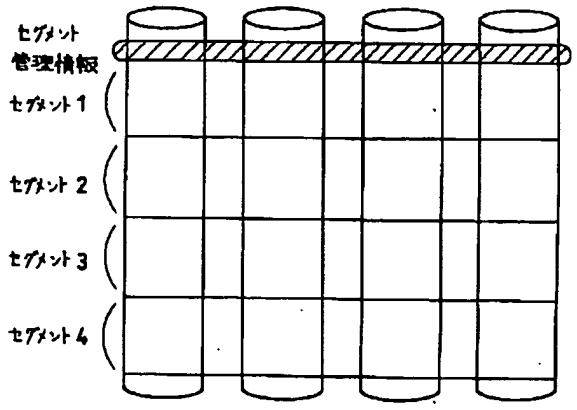
【図7】



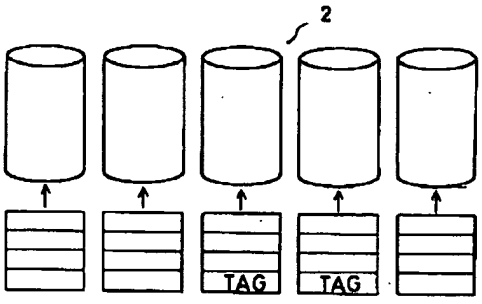
【図9】



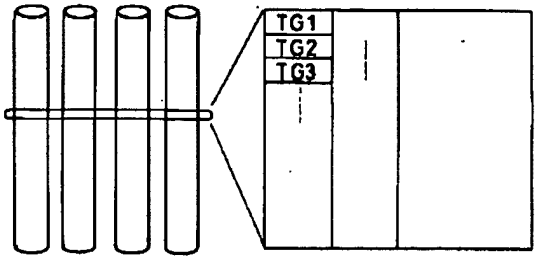
【図10】



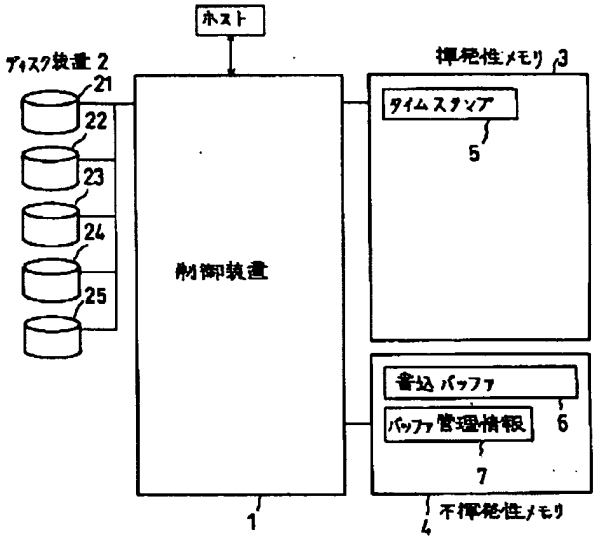
【図15】



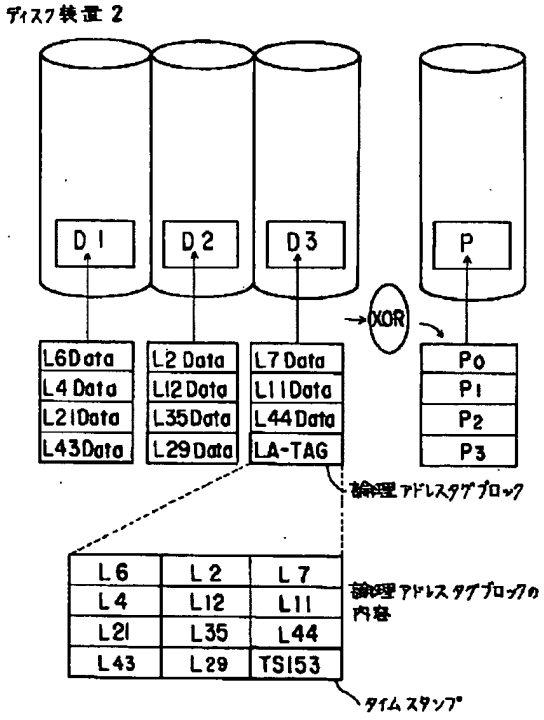
【図１２】



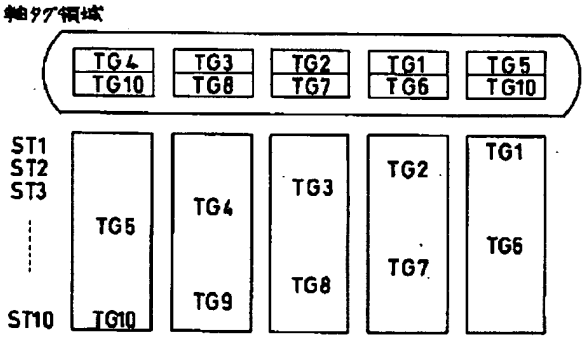
【図１３】



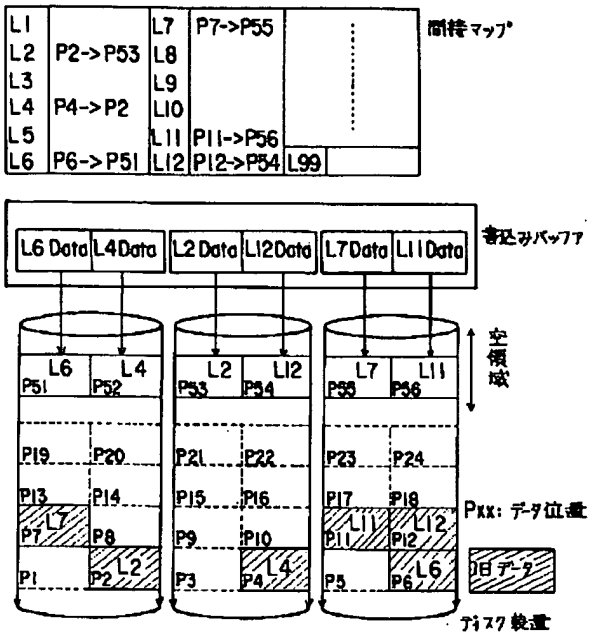
【図１４】



【図１６】



【図１７】


L1
L2 P2->P53
L3
L4 P4->P2
L5
L6 P6->P51
L7 P7->P55
L8
L9
L10
L11 P11->P56
L12 P12->P54 L99
間接マップ
L6 Data L4 Data L2 Data L12 Data L7 Data L11 Data
書込みバッファ
空領域
Pxx: データ位置
旧データ
ディスク装置